

Variable Naming

Allowed

`let userName`

Best Practice:
camelCase

`let ageGroup5`

Only letters
and digits

`let $kindOfSpecial`

Starting with \$
is allowed

`let _internalValue`

Starting with _
is allowed

Not Allowed / Not Recommended

`let user_name`

Allowed but
bad practice!

`let 21Players`

Starting digits
not allowed

`let user-b`

No special
characters!

`let let`

Keywords not
allowed



Variables & Constants

```
let userName = 'Max';
```

```
userName = 'Manu';
```

A “data container” / “data storage”

...where the value can change!

```
const totalUsers = 15;
```

```
totalUsers = 20;
```



A “data container” / “data storage”

...where the value must not change!



Use **constants** as often as possible (i.e. whenever you actually got data that never changes) to be clear about your intentions (in your code).

Numbers

2, -3, 22.956

Important for calculations and code where you need to “work with a number”

Strings (Text)

'Hi', "Hi", `Hi`

Important for outputting results, gathering input

Booleans

true / false

Important for conditional code and situations where you only have 2 options

Objects

{ name: 'Max',
age: 31 }

Important for grouped/ related data, helps you with organizing data

Arrays

[1, 3, 5]

Important for list data, unknown amounts of data



Functions

"Code on Demand"

(1) Define Function

```
function greetUser(name) {  
  alert('Hi ' + name);  
}
```

You can (but don't have to) use *parameters* (e.g. *name*) and you can (but don't have to) return values (via *return*).



(2) Call Function

```
greetUser('Max');
```

As often as you want, passing in (different) parameter values!

Every function execution then runs independent from (possible) other executions.



2x



3:02 / 5:50





null / undefined / NaN

Special Values

undefined

Default value of uninitialized variables

You shouldn't assign undefined as a value manually

null

Never assumed by default

You can assign this is a value if you want to "reset" / "clear" a variable

NaN

Not a type!

Technically, it's of type number and can therefore be used in calculations

It yields a new NaN and it's the result of invalid calculations (e.g. $3 * 'hi'$)

NOT entirely equal!



2x



3:56 / 6:20



Timeline Summary

